

Index

List of Practicals

Setting up CORDOVA, PhoneGap Project and environment.

1.	<ul style="list-style-type: none">• Creating and building simple “Hello World” App using Cordova• Adding and Using Buttons• Adding and Using Event Listeners
2.	<ul style="list-style-type: none">• Creating and Using Functions• Using Events• Handling and Using Back Button
3.	<ul style="list-style-type: none">• Installing and Using Plugins• Installing and Using Battery Plugin• Installing and Using Camera Plugin
4.	<ul style="list-style-type: none">• Installing and Using Device Plugin• Installing and Using Accelerometer Plugin
5.	<ul style="list-style-type: none">• Create and Using Prompt Function
6.	<ul style="list-style-type: none">• Installing and Using Network Information Plugin• Installing and Using Vibration Plugin
7.	<ul style="list-style-type: none">• Installing and Using Contacts Plugin

Practical 1A:

- Creating and building simple “Hello World” App using Cordova

```
C:\Mobile_Programming>cordova create HelloWorld io.c.hw HelloApp  
Creating a new cordova project.
```

```
C:\Mobile_Programming>
```

```
C:\Mobile_Programming>cd HelloWorld
```

```
C:\Mobile_Programming\HelloWorld>cordova platform add android
```

Then go to <<Your Folder>>\HelloWorld\www directory and open **index.html** file
add the code below <div class="app" >

```
<div class="app">
```

```
<h1> Hello world </h1>
```

```
<h2> Welcome to SYIT Mobile Programming</h2>
```

```
<h1> My Name is Abcd Roll No . 123 </h1>
```

Save the file

```
C:\Mobile_Programming\HelloWorld>cordova build android _
```

this will create a apk file at location given below

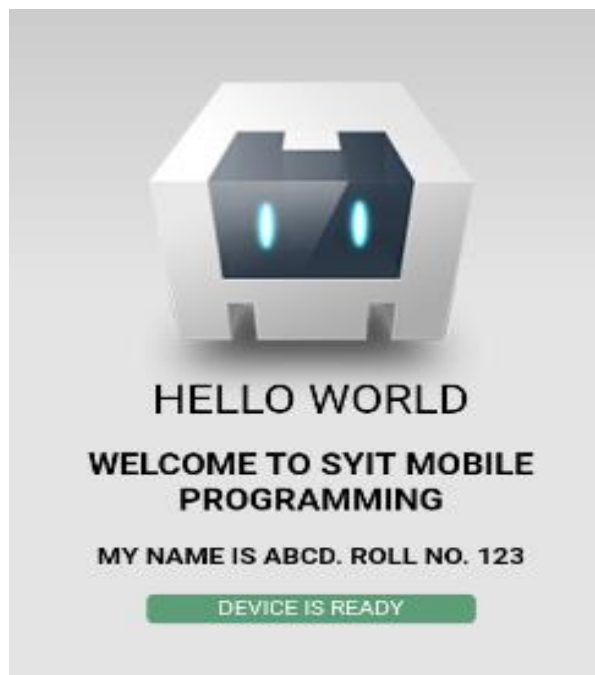
```
BUILD SUCCESSFUL in 1s  
47 actionable tasks: 1 executed, 46 up-to-date  
Built the following apk(s):  
  C:\Mobile_Programming\HelloWorld\platforms\android\app\build\outputs\apk\debug\app-debug.apk  
C:\Mobile_Programming\HelloWorld>
```

Rename this apk file to Prac1A.apk

Copy this app on your mobile device

install and test the app.

Output Practical 1A:



Practical 1B & 1C

Adding and Using Buttons

Create a button called "C language". when you click on the button it should provide the information of C language Functions.

Create a button called "Bank". when you click on the button it should provide the information of types of accounts.

```
C:\Mobile_Programming>cordova create Bank io.c.b BankApp
```

```
C:\Mobile_Programming>cd Bank
```

```
C:\Mobile_Programming\Bank>cordova platform add android
```

```
Goto C:\Mobile_Programming\Bank\www folder and edit index.html
```

after `<div class="app">`

```
<button id="cInfo" > C language </button><br>
```

```
<button id="btnBank" >Bank</button>
```

before `<h1>Apache Cordova</h1>`

save the index.html file

Then go to js folder Bank\www\js folder and open the index.js file using notepad as follows

add the following code **inside function onDeviceReady() {**

```
document.getElementById("cInfo").addEventListener("click", cFun );
```

```
document.getElementById("btnBank").addEventListener("click", bankFun );
```

then add the following code after function **onDeviceReady() { };**

```
function cFun(){
```

```
    alert("C is a general-purpose, procedural computer programming language supporting structured programming
```

```
");
```

```
};
```

```
function bankFun(){
```

```
    alert(  "\n1. Saving Account."+
```

```
        "\n2. Regular Savings."+
```

```
        "\n3. Current Account."+
```

```
        "\n4. Recurring Deposit Account."+
```

```
        "\n5. Fixed Deposit Account."+
```

```
        "\n6. DEMAT Account."+
```

```
        "\n7. NRI Accounts.");
```

```
};
```

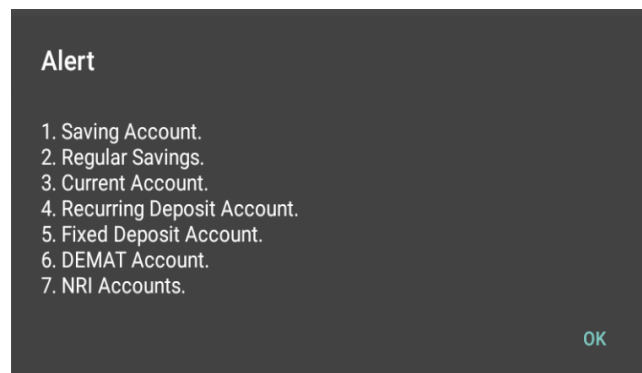
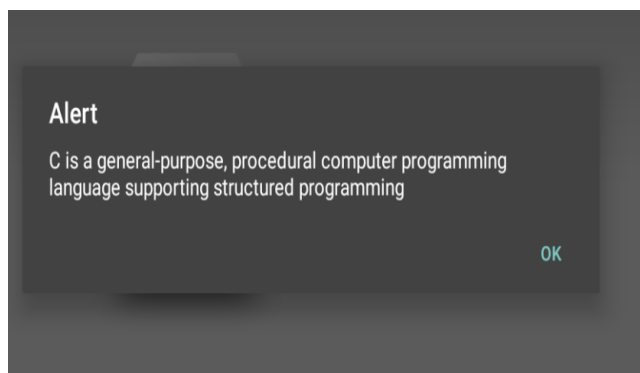
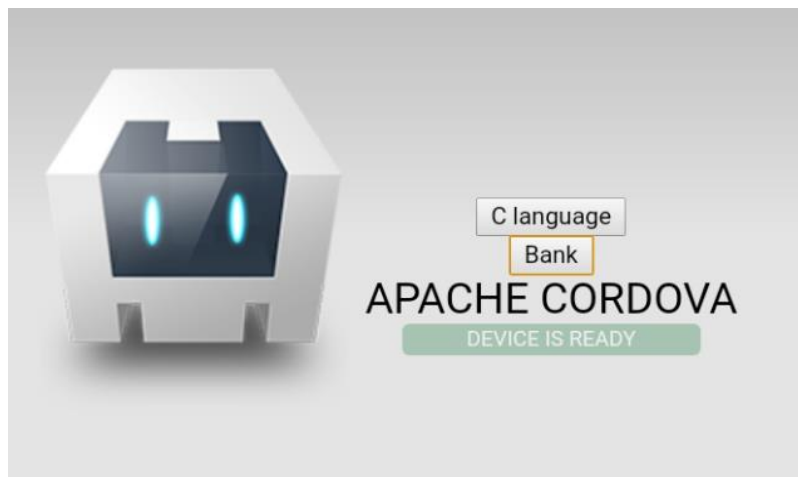
save the file

build the app go to Bank\platforms\android\app\build\outputs\apk\debug\app-debug.apk and rename it as

BankApp

copy on your mobile device and test.

Output Practical 1B&C:



Practical No . 2

- Adding and Using Event Listeners
- Creating and Using Functions
- Handling and Using Back Button

```
Mobile_Programming> cordova create EventDemo io.c.ed EventApp
```

```
Mobile_Programming> cd EventDemo
```

```
Mobile_Programming\EventDemo> cordova platform add android
```

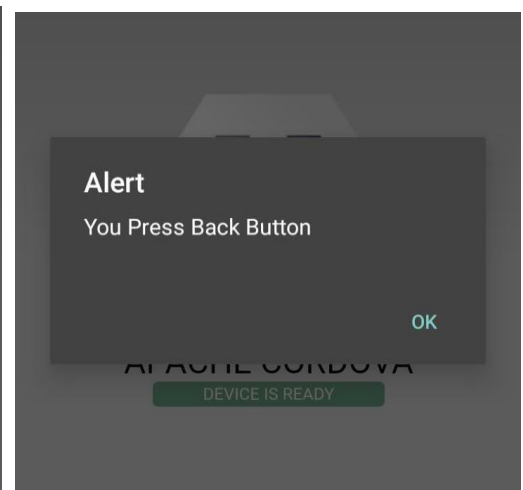
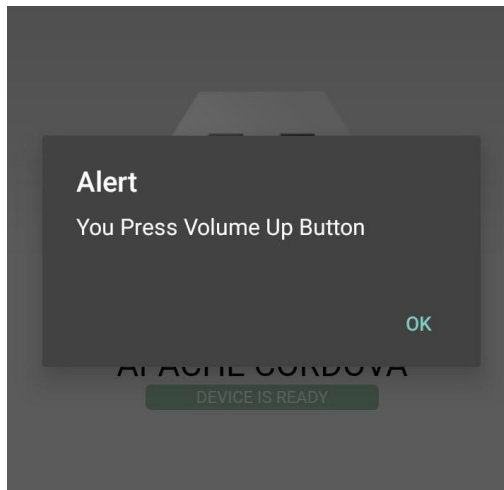
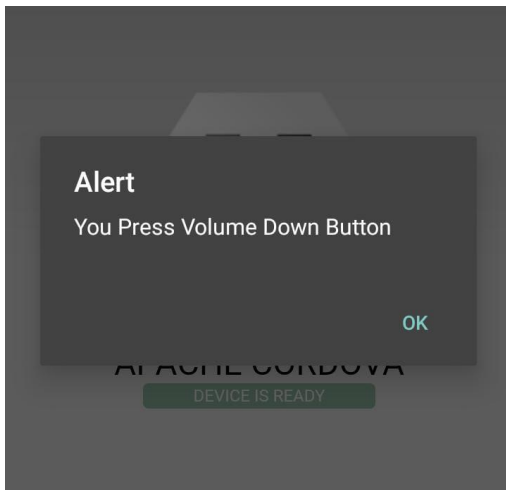
go to Mobile_Programming\EventDemo\www\js folder open index.js file add the code inside **function onDeviceReady() {**

```
document.addEventListener("volumeupbutton", upFun, false);
document.addEventListener("volumedownbutton", dwFun, false);
document.addEventListener("backbutton", bkFun, false);
```

and in the same file add the following code before **app.initialize();**

```
function upFun(){
    alert("You Press Volume Up Button ");
};
function dwFun(){
    alert("You Press Volume Down Button" );
};
function bkFun(e){
    e.preventDefault(); // Will not close the App
    alert("You Press Back Button");
};
```

Output Practical 2:



Practical 3A: Installing and using plugin

C:\Mobile_Programming> **npm install -g plugman**

to install plugman to install and configure various plugins to be used in cordova project.

plugins can be later added to a program using command

cordova plugin add <<name of plugin>>

Practical 3B: Installing and Using Battery Plugin

Administrator: Mobile

```
C:\Mobile_Programming>cordova create BatteryDemo io.c.bat BatApp
Creating a new cordova project.
```

```
C:\Mobile_Programming>cd BatteryDemo
```

```
C:\Mobile_Programming\BatteryDemo>cordova platform add android
```

after this go to C:\Mobile_Programming\BatteryDemo\www\js folder and open index.js file in editor

add the following code after **function onDeviceReady() {**

```
window.addEventListener("batterystatus", batStatFun, false);
window.addEventListener("batterylow", batLowFun, false);
window.addEventListener("batterycritical", batBatCr, false);
```

in the same file add the following code **before app.initialize();**

```
function batStatFun(info){
    alert("BATTERY STATUS "+
        "\n Level is "+info.level+
        "\n Is Charger Connected "+info.isPlugged
    );
};
```

```
function batLowFun(status){
    alert(" Battery Low !!!"+
        status.level+" % Remaining"+
        "\n Charge your Phone !!!! " );
};
```

```
function batBatCr(status){
    alert("Critical Condition \n Battery about to die"+
        status.level+"% charging Left \n Hurry Up");
};
```

save the file

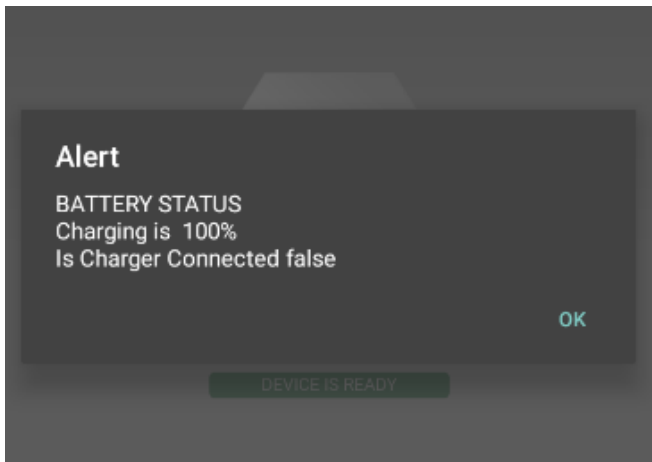
add battery plugin to project

```
C:\Mobile_Programming\BatteryDemo>cordova plugin add cordova-plugin-battery-status
```

build the app and test.

```
C:\Mobile_Programming\BatteryDemo>cordova build android
```

Output Practical 3B:



Practical 3C:

- Installing and Using Camera Plugin

```
C:\Mobile_Programming>cordova create CamDemo io.c.cam CamApp
```

```
C:\Mobile_Programming>cd CamDemo
```

```
C:\Mobile_Programming\CamDemo>cordova platform add android
```

```
C:\Mobile_Programming\CamDemo>cordova plugin add cordova-plugin-camera
```

go to **Mobile_Programming\CamDemo\www** and open **index.html** file in editor and the following code after
<div class="app">

```
<button id = "takePic">TAKE PICTURE</button>  
<img id = "myPhoto" height="200" width="200" ></img>
```

then go to **Mobile_Programming\CamDemo\www\js** and open **index.js** file in editor and the following code after

```
onDeviceReady: function() {
```

```
document.getElementById("takePic").addEventListener  
("click", camFun);
```

then add the following code **before** **app.initialize();**

```
function camFun() {  
    navigator.camera.getPicture( onSuccess, onFail, {  
        quality: 50,  
        destinationType: Camera.DestinationType.DATA_URL  
    } );
```

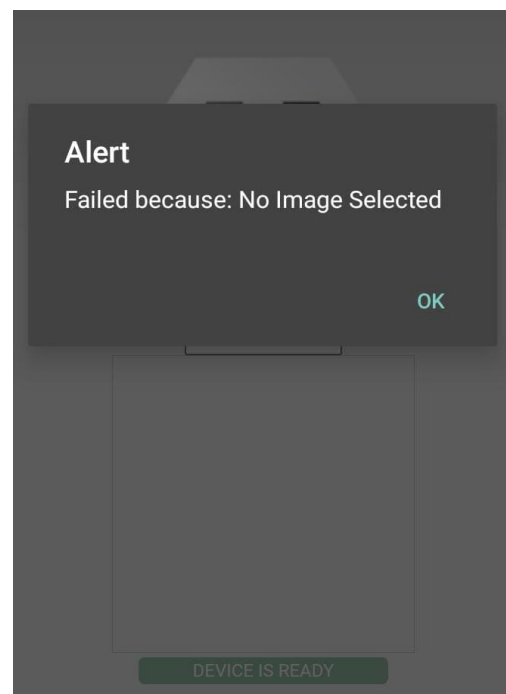
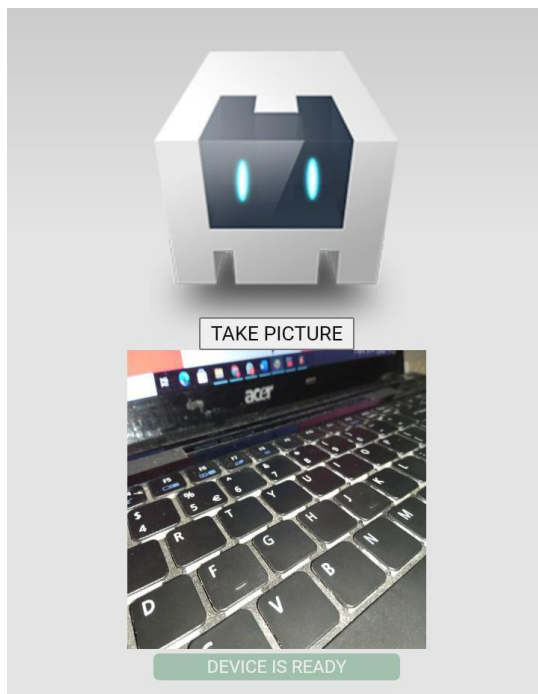
```
function onSuccess(imageData) {  
    var image = document.getElementById('myPhoto');  
    image.src = "data:image/jpeg;base64," + imageData;  
}
```

```
function onFail(message) {  
    alert('Failed because: ' + message);  
}  
};
```

```
C:\Mobile_Programming\CamDemo>cordova build android
```

Copy the program on your mobile phone, install and test. click on Take **Picture button** of your App if camera starts, take a picture and select **OK**.

Output Practical 3C:



Practical 4A:

- Installing and Using Device Plugin

```
C:\Mobile_Programming>cordova build DivDemo io.cor.dd DivApp_
```

```
C:\Mobile_Programming>cd DivDemo
```

```
C:\Mobile_Programming\DivDemo>cordova platform add android
```

```
C:\Mobile_Programming\DivDemo>cordova plugin add cordova-plugin-device  
Plugin "cordova-plugin-device" already installed on android.
```

```
Adding cordova-plugin-device to package.json  
Saved plugin info for "cordova-plugin-device" to config.xml
```

open index.html file from **www** folder and add the line below **<div class="app">**

```
<button id = "btnDiv" >KNOW YOUR DEVICE</button>
```

open index.js file from **js** folder and add the line below **onDeviceReady: function() {**

```
document.getElementById("btnDiv").addEventListener("click", divFun);
```

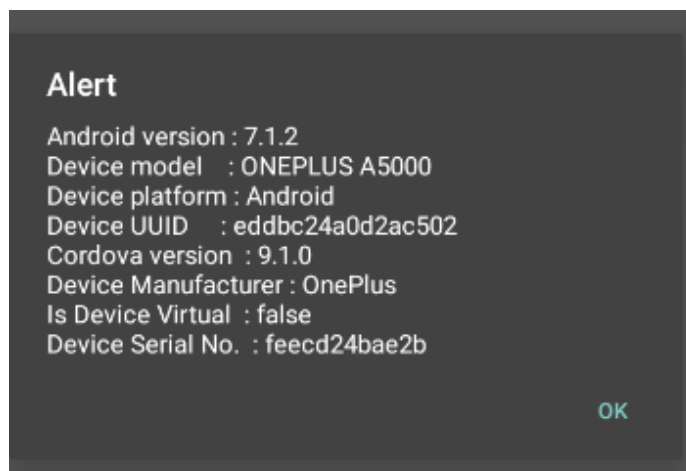
and before **app.initialize();**

```
function divFun(){  
alert("Android version : " + device.version + "\n" +  
    "Device model   : " + device.model + "\n" +  
    "Device platform : " + device.platform + "\n" +  
    "Device UUID    : " + device.uuid + "\n" +  
    "Cordova version : " + device.cordova + "\n" +  
    "Device Manufacturer: " + device.manufacturer+ "\n" +  
    "Is Device Virtual : " + device.isVirtual + "\n" +  
    "Device Serial No. : " + device.serial);  
};
```

```
C:\Mobile_Programming\DivDemo>cordova build android_
```

copy app on mobile phone and test the app.

Output Practical 4A:



Practical No. 4B

- **Installing and Using Accelerometer Plugin**

```
C:\Mobile_Programming>cordova create OriDemo io.cor.lib OriApp  
Creating a new cordova project.
```

```
C:\Mobile_Programming>cd OriDemo
```

```
C:\Mobile_Programming\OriDemo>cordova platform add android
```

```
C:\Mobile_Programming\OriDemo>cordova plugin add cordova-plugin-device-motion
```

goto www folder and open index.html file and add the code below <div class="app">

```
<button id = "getAcc">GET ACCELERATION</button>
```

```
<br><br>
```

```
<button id = "watchAcc">WATCH ACCELERATION</button>
```

now open index.js from js folder and add the code below onDeviceReady: function() {

```
document.getElementById("getAcc").addEventListener("click", getAccFun);
```

```
document.getElementById("watchAcc").addEventListener("click", watchAccFun);
```

and the following code before app.initialize();

```
function getAccFun() {
```

```
    navigator.accelerometer.getCurrentAcceleration(  
        accelerometerSuccess, accelerometerError);
```

```
    function accelerometerSuccess(acceleration) {  
        alert('Acceleration X: ' + acceleration.x + '\n' +  
            'Acceleration Y: ' + acceleration.y + '\n' +  
            'Acceleration Z: ' + acceleration.z + '\n' +  
            'Timestamp: ' + acceleration.timestamp +
```

```
        '\n');  
    };
```

```
    function accelerometerError() {  
        alert('onError!');  
    };
```

```
};
```

```
function watchAccFun() {  
    var accelerometerOptions = {  
        frequency: 3000  
    }  
    var watchID =
```

```
navigator.accelerometer.watchAcceleration(  
    accelerometerSuccess, accelerometerError,
```

```
    accelerometerOptions);
```

```
    function accelerometerSuccess(acceleration) {
```

```

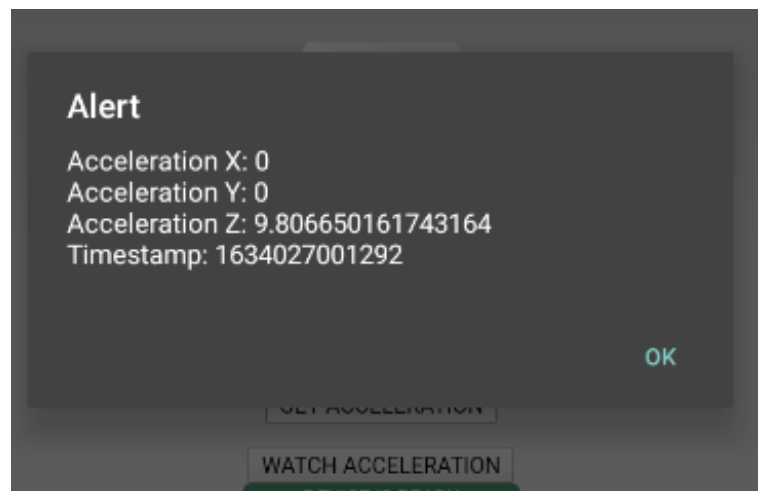
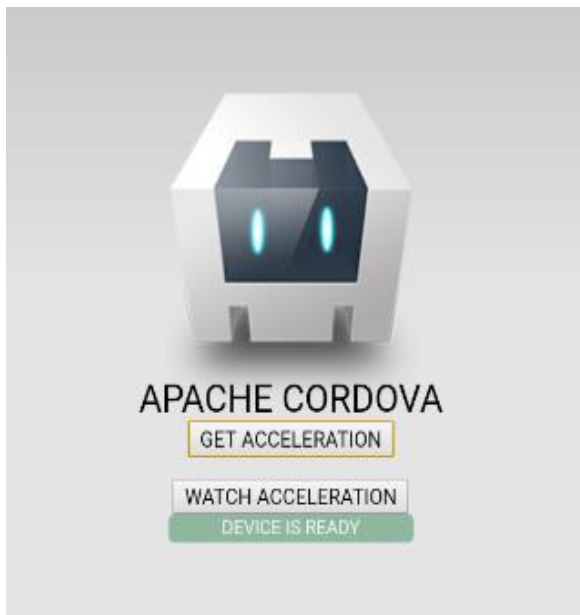
alert('Acceleration X: ' + acceleration.x + '\n' +
'Acceleration Y: ' + acceleration.y + '\n' +
'Acceleration Z: ' + acceleration.z + '\n' +
'Timestamp: ' + acceleration.timestamp +
'\n');

setTimeout(function() {
    navigator.accelerometer.clearWatch(watchID);
}, 10000);
};

function accelerometerError() {
    alert('onError!');
};
};
save both the files and build using
C:\Mobile_Programming\OriDemo>cordova build android

```

Output Practical 4B:



Practical No. 5

• Create and Using Prompt Function

```
C:\Mobile_Programming>cordova create DialogDemo cor.io.dd DialogApp  
Creating a new cordova project.
```

```
C:\Mobile_Programming>cd DialogDemo
```

```
C:\Mobile_Programming\DialogDemo>cordova platform add android
```

```
C:\Mobile_Programming\DialogDemo>cordova plugin add cordova-plugin-dialogs  
now open index.html from www folder and add the following code after <div class="app">
```

```
<button id = "dialogAlert">ALERT </button> <br>  
<button id = "dialogConfirm">CONFIRM</button> <br>  
<button id = "dialogPrompt">PROMPT</button> <br>  
<button id = "dialogBeep">BEEP</button> <br>
```

now open index.js file from js folder and add the following code below onDeviceReady: function() {

```
document.getElementById("dialogAlert").addEventListener("click", AlertFun);  
document.getElementById("dialogConfirm").addEventListener("click", ConfirmFun);  
document.getElementById("dialogPrompt").addEventListener("click", PromptFun);  
document.getElementById("dialogBeep").addEventListener("click", BeepFun);
```

and the following code before app.initialize();

```
function AlertFun() {  
    var message = "Dont use phone while on Charging !!";  
    var title = "ALERT";  
    var buttonName = "Alert Button";  
    navigator.notification.alert(message, alertCallback, title, buttonName);  
  
    function alertCallback() { alert("Alert is Dismissed!"); }  
};  
  
function ConfirmFun() {  
    var message = "Do You Really want to Exit?";  
    var title = "CONFIRM";  
    var buttonLabels = "YES,NO";  
    navigator.notification.confirm(message, confirmCallback, title, buttonLabels);  
  
    function confirmCallback(buttonIndex) { alert("Ok Bye.... "); }  
};  
  
function PromptFun() {  
    var message = "Enter Your Name?";  
    var title = "PROMPT";  
    var buttonLabels = ["YES", "NO"];  
    var defaultText = "ABCD";  
    navigator.notification.prompt(message, promptCallback,  
        title, buttonLabels, defaultText);  
  
    function promptCallback(result) { alert("Your Input is : " + result.input1); }  
};
```

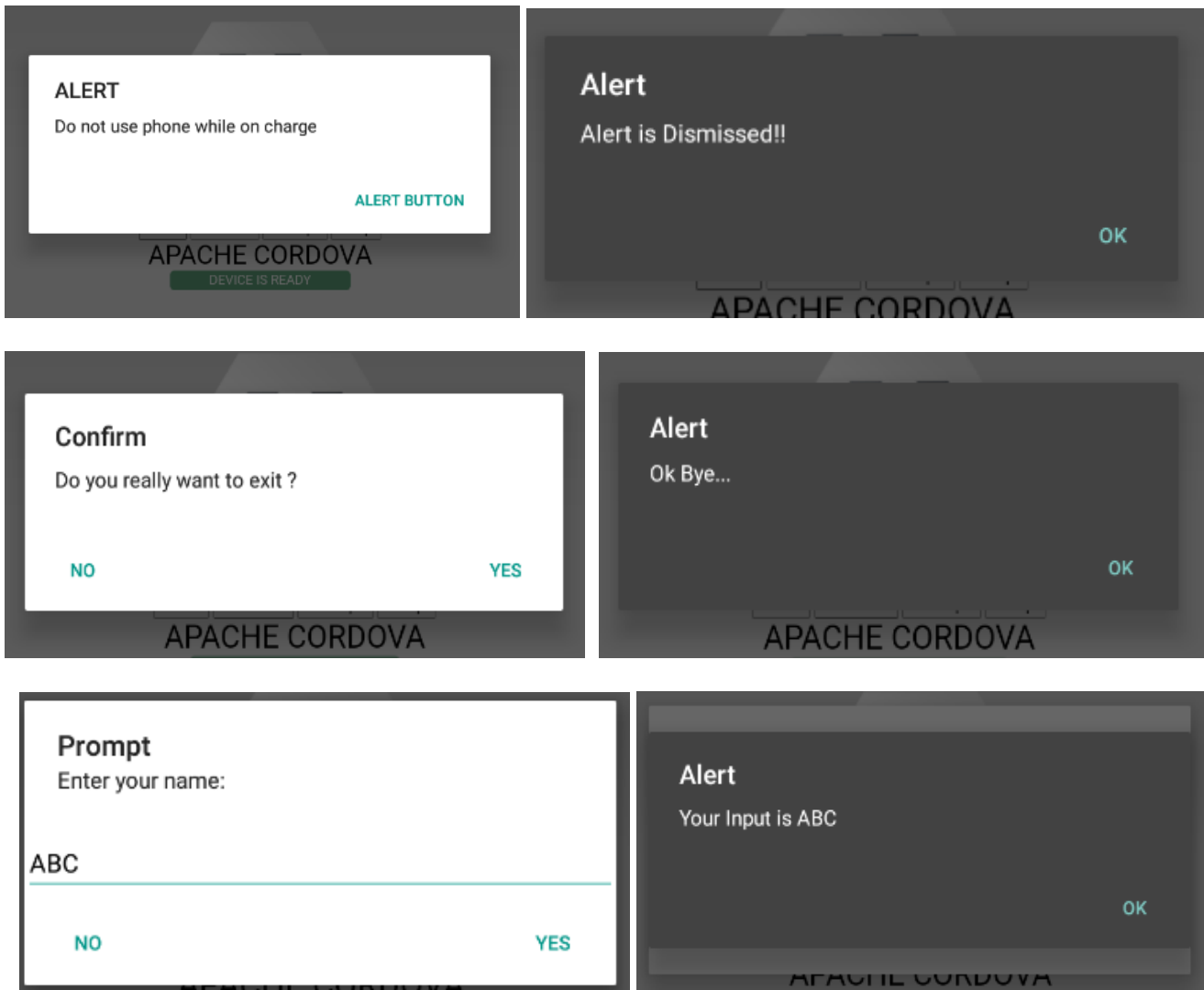
```
};
```

```
function BeepFun() { navigator.notification.beep(5); //beep(times) times is number of beeps  
};
```

C:\Mobile_Programming\DialogDemo>cordova build android

copy the app on your mobile device and test the App.

OUTPUT PRACTICAL 5:



Practical No. 6A

Installing and Using Network Information Plugin

```
C:\Mobile_Programming>cordova create NetDemo io.cor.lib NetApp  
Creating a new cordova project.
```

```
C:\Mobile_Programming>cd NetDemo
```

```
C:\Mobile_Programming\NetDemo>cordova platform add android
```

```
C:\Mobile_Programming\NetDemo>cordova plugin add cordova-plugin-network-information
```

Go to www folder and open index.html to add the code below <div class="app">

```
<button id = "btnInfo">NETWORK INFORMATION</button>
```

now go to index.js file in js folder and add the following code below onDeviceReady: function() {

```
document.getElementById("btnInfo").addEventListener("click", NetFun);  
document.addEventListener("offline", offFun, false);  
document.addEventListener("online", onFun, false);
```

and the following before app.initialize();

```
function NetFun(){  
var networkState = navigator.connection.type;  
var states = { };  
states[Connection.UNKNOWN] = 'Unknown connection';  
states[Connection.ETHERNET] = 'Ethernet connection';  
states[Connection.WIFI] = 'WiFi connection';  
states[Connection.CELL_2G] = 'Cell 2G connection';  
states[Connection.CELL_3G] = 'Cell 3G connection';  
states[Connection.CELL_4G] = 'Cell 4G connection';  
states[Connection.NONE] = 'No network connection';
```

```
alert('Connection type: ' + states[networkState]);
```

```
};
```

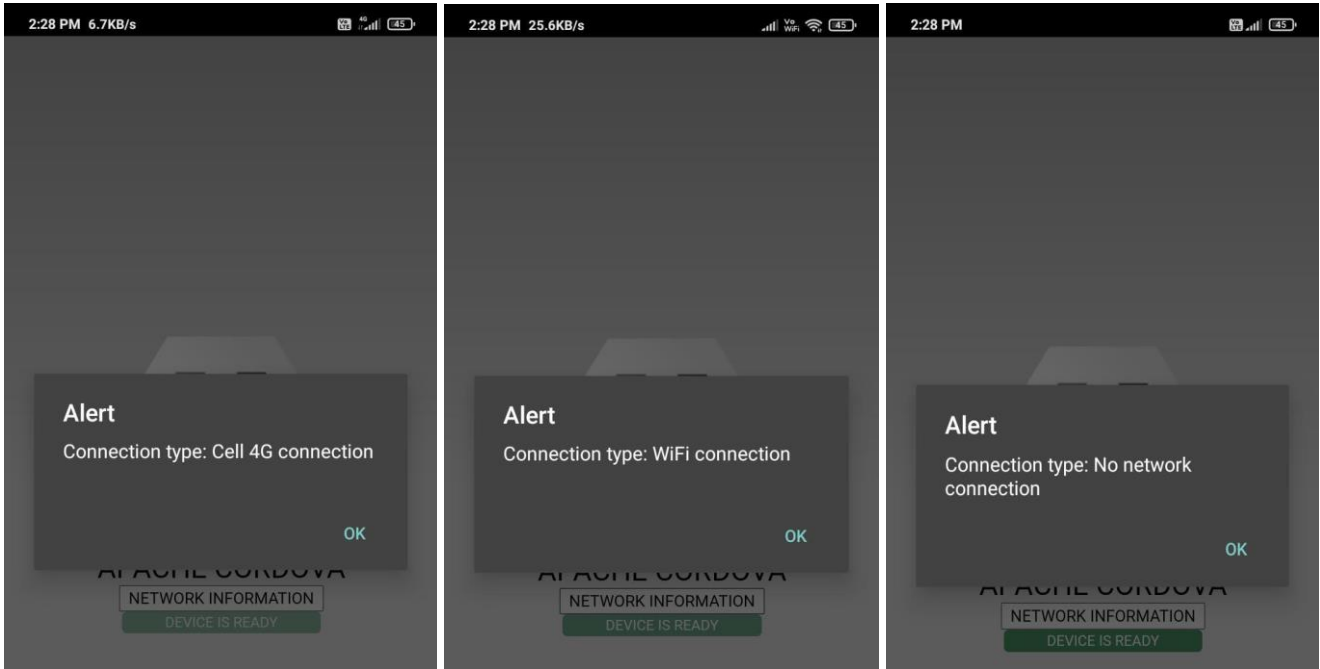
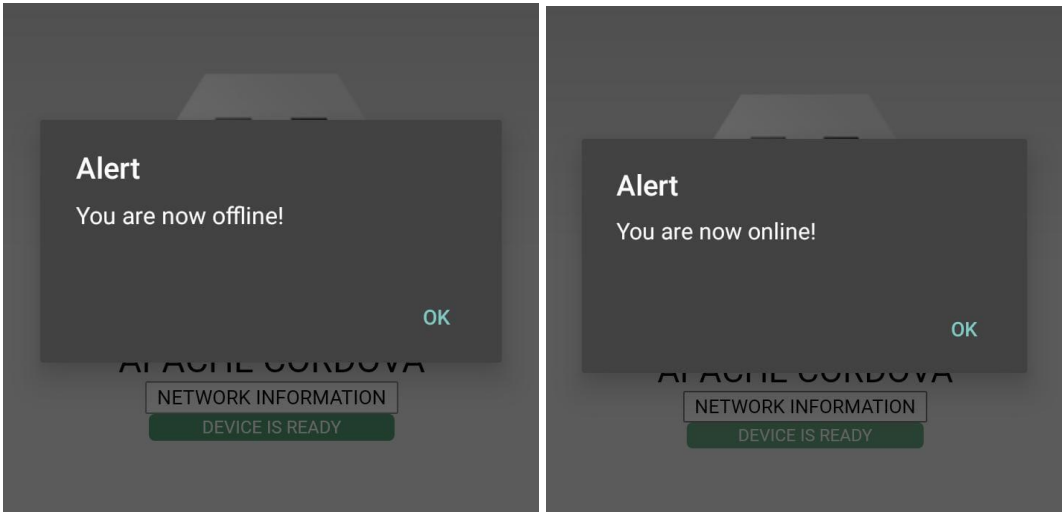
```
function offFun(){  
alert('You are now offline!');  
};
```

```
function onFun(){  
alert('You are now online!');  
};
```

```
C:\Mobile_Programming\NetDemo>cordova build android
```

copy the program and try different network modes by enabling and disabling mobile data network or wifi .

Output Practical 6A:



Practical No. 6B : Installing and Using Vibration Plugin

1. Creating Project Folder name as Vibration and App Name as VibrationApp

```
WIN-10@DESKTOP-4VN1SRA MINGW64 ~/Desktop/App/Outut App All
$ cordova create Vibration com.example.com VibrattionApp
Creating a new cordova project.
```

2. Go to Vibration Folder and Add Platform Android

```
WIN-10@DESKTOP-4VN1SRA MINGW64 ~/Desktop/App/Outut App All
$ cd Vibration

WIN-10@DESKTOP-4VN1SRA MINGW64 ~/Desktop/App/Outut App All/Vibration
$ cordova platform add android
```

3. Add cordova-plugin-vibration

```
WIN-10@DESKTOP-4VN1SRA MINGW64 ~/Desktop/App/Outut App All/Vibration
$ cordova plugin add cordova-plugin-vibration
Installing "cordova-plugin-vibration" for android
Adding cordova-plugin-vibration to package.json
```

4. Go to www folder and open index.html to add the code below <div class="app">

<button id="vib">Click to Vibrate</button>

5. Go to js folder and open index.js and add event listener code inside **function onDeviceReady()**{

document.getElementById('vib').addEventListener('click', VibrateFun);

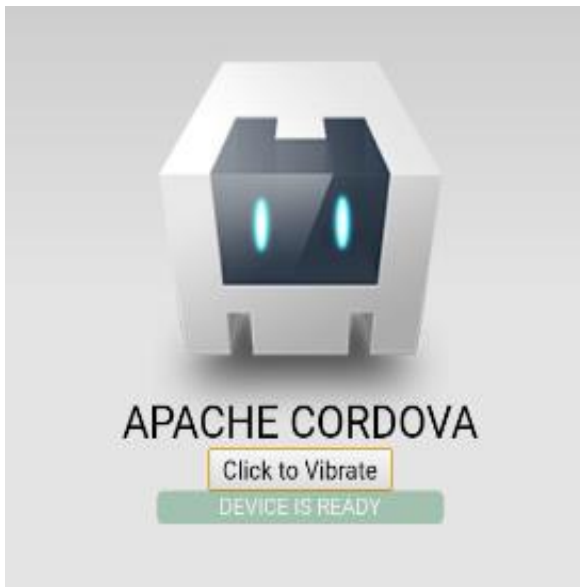
6. Add Function VibrateFun at ending of the file

```
function VibrateFun(){
    navigator.vibrate(5000); // navigator.vibrate(5000) means 5 sec vibration
};
```

7. Build Android and Install & Test App

```
WIN-10@DESKTOP-4VN1SRA MINGW64 ~/Desktop/App/Outut App All/Vibration
$ cordova build android
BUILD SUCCESSFUL in 54s
40 actionable tasks: 40 executed
Built the following apk(s):
   C:\Users\WIN-10\Desktop\App\Outut App All\Vibration\platforms\android\
app\build\outputs\apk\debug\app-debug.apk
```

Output Practical 6B:



Practical No. 7 : Installing and Using Contacts Plugin

1. Creating Project Folder name as Contact and App Name ad ContactApp

```
WIN-10@DESKTOP-4VN1SRA MINGW64 ~/Desktop/App/Outut App All
$ cordova create Contact com.example.contact ContactApp
Creating a new cordova project.
```

2. Go to Contact Folder and Add Platform Android

```
WIN-10@DESKTOP-4VN1SRA MINGW64 ~/Desktop/App/Outut App All
$ cd Contact

WIN-10@DESKTOP-4VN1SRA MINGW64 ~/Desktop/App/Outut App All/Contact
$ cordova platform add android
```

3. Add Plugin Associated with contact

```
WIN-10@DESKTOP-4VN1SRA MINGW64 ~/Desktop/App/Outut App All/Contact
$ cordova plugin add https://github.com/apache/cordova-plugin-contacts.git
Adding cordova-plugin-contacts to package.json
```

4. Go to www folder and open index.html to add the code below <div class="app">

```
<button id="save">Add Contact</button>
<button id="find">Find Contact</button>
<button id="delete">Delete Contact</button>
```

5. Go to js folder and open index.js and add event lister code inside **function onDeviceReady(){**

```
document.getElementById('save').addEventListener('click', getContact);
document.getElementById('find').addEventListener('click', findContact);
document.getElementById('delete').addEventListener('click', deleteCon);
```

6. Add Function Body at the bottom of the file

```
function getContact(){
    var myContact = navigator.contacts.create({ "displayName": "Cordova Contact" });
    var phoneNumbers = [];
    phoneNumbers[0] = new ContactField('work', '8840926059', false);
    phoneNumbers[1] = new ContactField('mobile', '7304121104', true); // preferred number
    myContact.phoneNumbers = phoneNumbers;
    myContact.save(onSuccess, onError);
};
function onSuccess(myContact) {
    alert("Save Success");
}
```

```

};

function onError(myContactError) {
    alert("Error = " + myContactError.code);
};

function findContact(){
    var options    = new ContactFindOptions();
    options.filter  = "Cordova Contact";
    options.multiple = true;
    options.desiredFields = [navigator.contacts.fieldType.id];
    options.hasPhoneNumber = true;
    var fields      = [navigator.contacts.fieldType.displayName,
        navigator.contacts.fieldType.name];
    navigator.contacts.find(fields, onfind, onError, options);
};

function onfind(contacts) {
    alert('Found ' + contacts.length + ' contacts.');
```

```

};

function deleteCon() {
    var options = new ContactFindOptions();
    options.filter = "Cordova Contact";
    options.multiple = false;
    fields = ["displayName"];
    navigator.contacts.find(fields, contactfindSuccess, contactfindError, options);

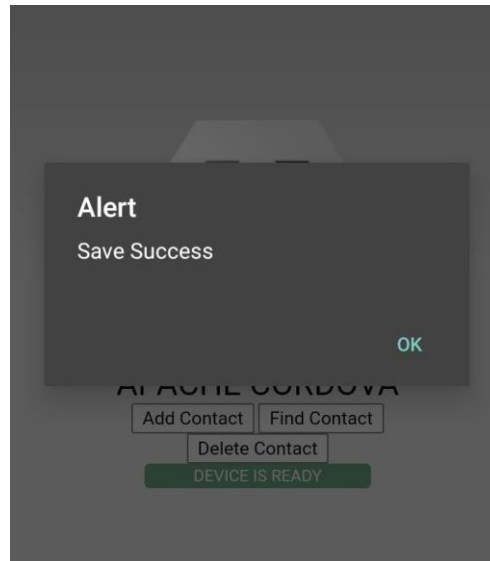
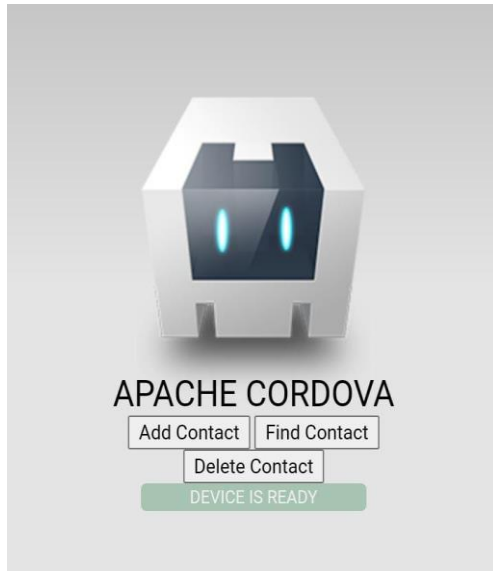
    function contactfindSuccess(contacts) {
        var contact = contacts[0];
        contact.remove(contactRemoveSuccess, contactRemoveError);

        function contactRemoveSuccess(contact) {
            alert("Contact Deleted");
        };
        function contactRemoveError(message) {
            alert('Failed because: ' + message);
        };
    };
    function contactfindError(message) {
        alert('Failed because: ' + message);
    };
};

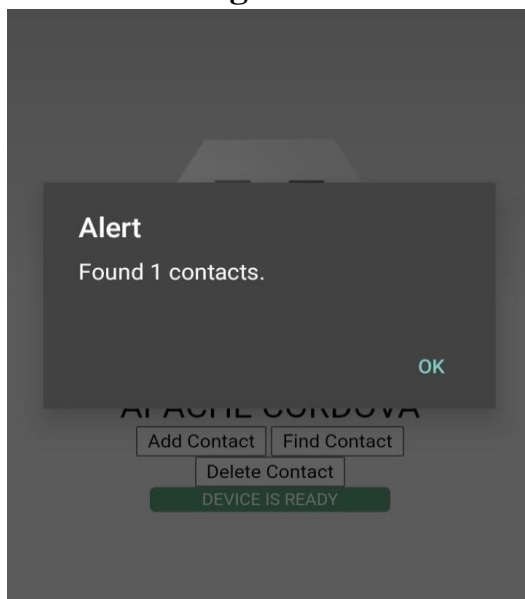
```

Output Practical 7:

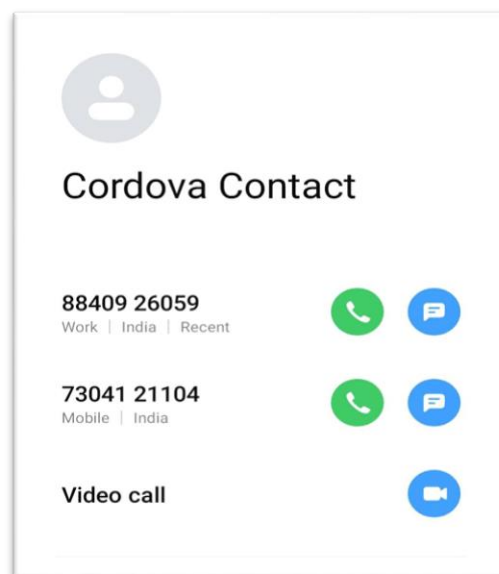
On Clicking Add Contact



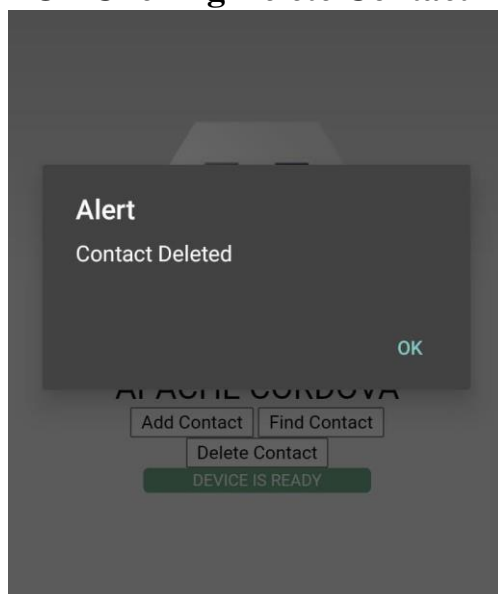
On Clicking Find Contact



Phone Contact List



On Clicking Delete Contact



No Contact Found

